Exoplanet Transit Detection using Deep Neural Networks

By Dinis Marques (P13240786)

Research Overview

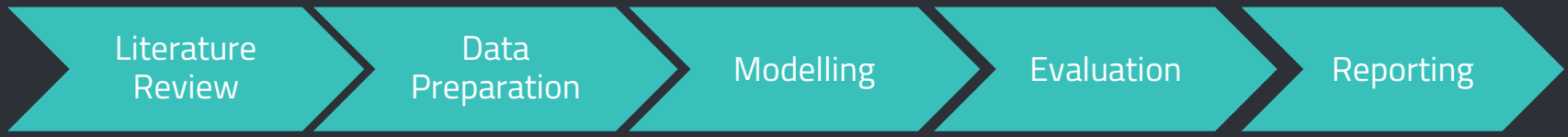
Motivation

- Space exploration benefits
- Big universe, big data
- New research frontier

Objectives

- Explore the possibility for the usage of deep learning architectures to solve problems in Astronomy, more specifically for the detection of exoplanets.
- Review relevant and recent literature and learn about suitable approaches.
- Find appropriate sources of telescope data, and develop methods for data retrieval and storage.
- Learn new methods to process the data to improve final model performance.
- Develop neural networks to successfully model data for the detection of exoplanets.
- Evaluate the effectiveness of the developed models.

Process



- Reveal appropriate problem domains in Astronomy
- Review popular deep learning architectures
- Learn how deep learning has been used to tackle problems in this field.
- Understand the current state of the art.

- Understanding
- Retrieval and storage
- Pre-processing

- Developing neural network models using Keras
- Hyperparameter optimization

- Cross validation
- Determining which metrics would be suitable for evaluation the models.
- Visualization of model performance using plots

- Dissertation
- Presentation

1 Literature Review

Overview

Architectures:

- MLP
- CNN
- LSTM

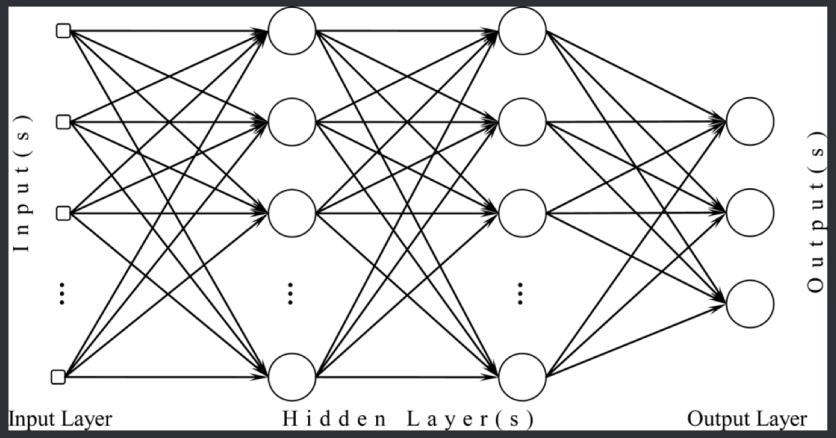
Popular applications in Astronomy:

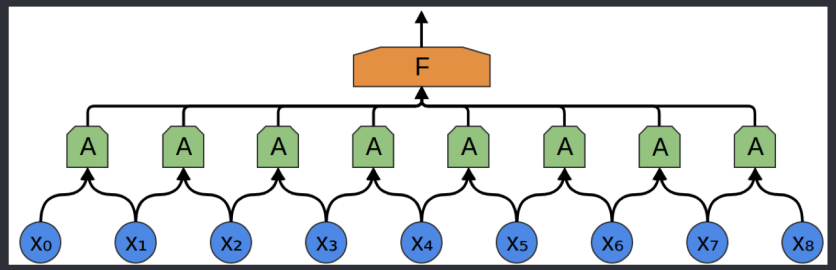
- Exoplanet Classification
- Supernovae Classification
- Gravitational Wave Classification

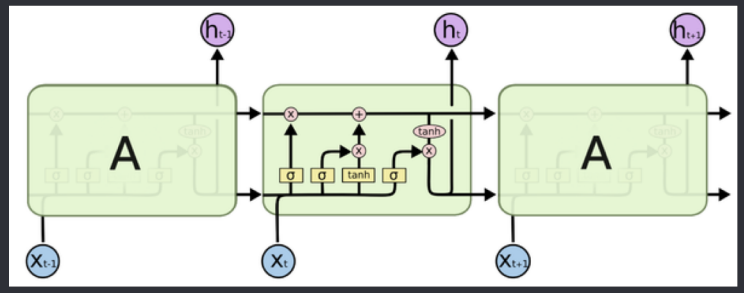
Type of problem domains:

- Time-series
- Image recognition

Architectures





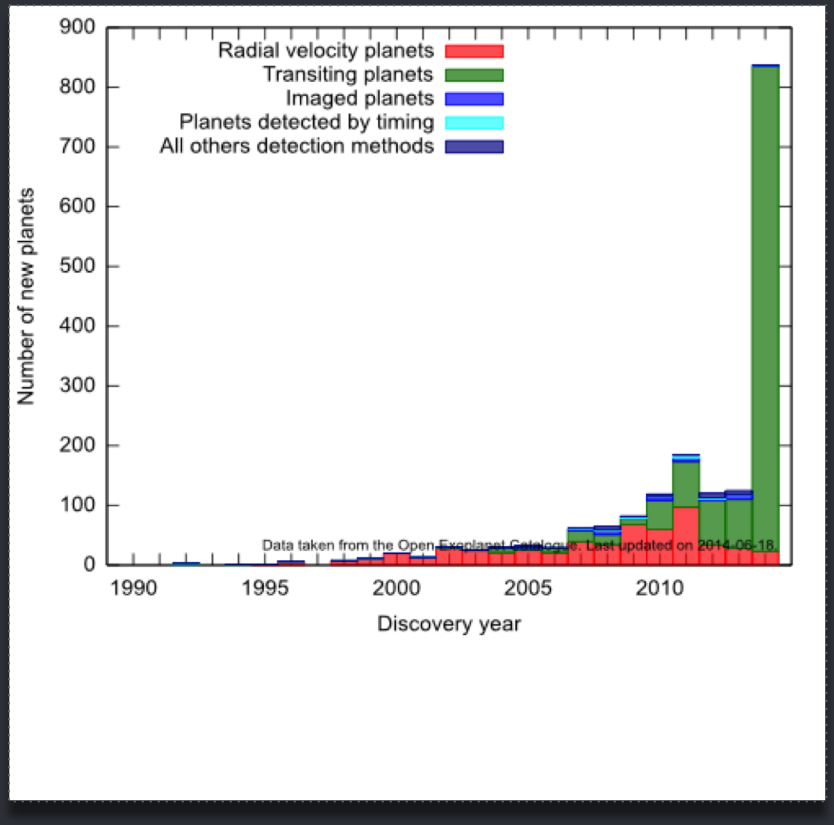


2 Data

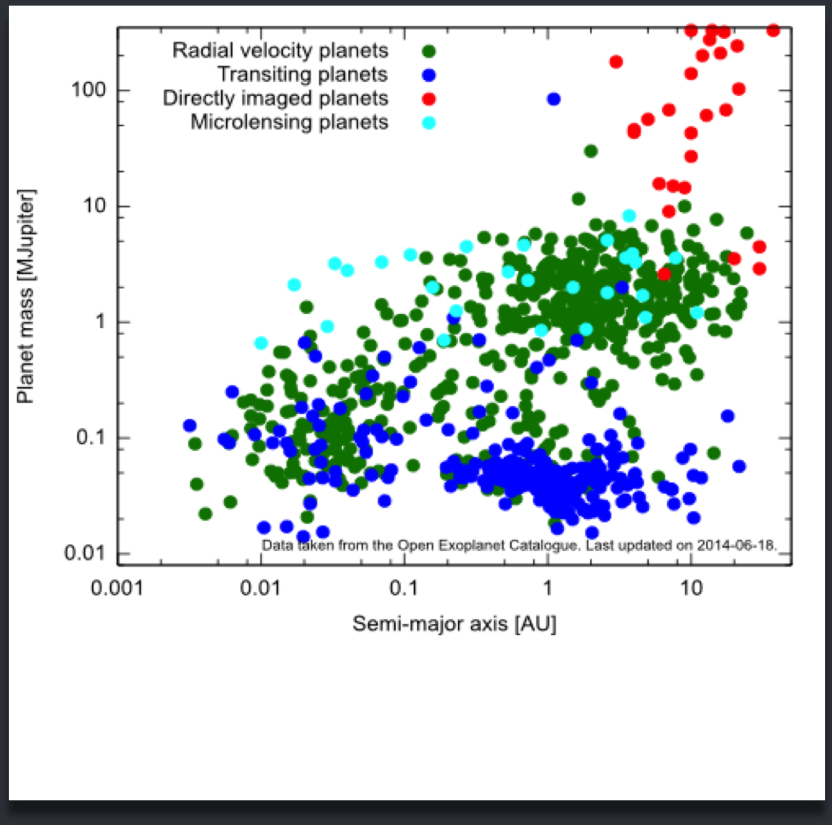
Kepler Mission

- Kepler and K2 telescope surveys
- Designed as statistical missions to:
  - Determine the presence of earth sized planets in the habitable zone around stars.
  - Determine the properties of such planets and parent stars.
- Observed a large portion of the sky almost continuously for 7 years and monitored the brightness of over 150,000 stars
  - Approximately **4000** transiting planets classified between the two missions.
  - Thousands more dispositioned as potential candidates.
    - Most detections in the past decade

Discovery Methods



Number Planets vs Discovery Year



Planet Mass vs Semi-Major axis

Kepler Pipeline

- Consists of several components
  - **CAL Module** - Raw data from the telescope sensors is converted into pixel data.
    - A timestamped pixel image for each observation.
    - Each observation lasts either 1 minute or 30 mins depending on the cadence.
  - **PA Module (Photometric Analysis)** - Uses simple aperture photometry (SAP) to convert the series pixel images into photometric light curves.
    - Flux values representative of the converted pixel information for each timestep
    - Multiple timely quarters of approx. 90 days
  - **PDC Module (Presearch Data Conditioning)** - Processes the SAP light curves to remove errors introduced by instruments and flux anomalies (cosmic rays, solar flares)
- Photometric data is stored at MAST.

Data

- Labelled data obtained from NASA Exoplanet Archive
  - Q1-Q17 DR24 catalogue - Used to train Autovetter
  - Data on approx. **15740** Threshold-Crossing Events (TCEs)
    - Transit depth, period, etc...
  - TCE labels - Planet Candidate (PC), Astronomical false positive (AFP), Non-Transiting phenomena (NTP).
  - Binary classification problem so AFP and NTP samples were merged into a single class.
- Retrieval of lightcurve data from MAST
  - TCE information obtained from table
    - WGET commands used to download data
- Total of **3600** positive samples, and **12137** negative samples

Analysis

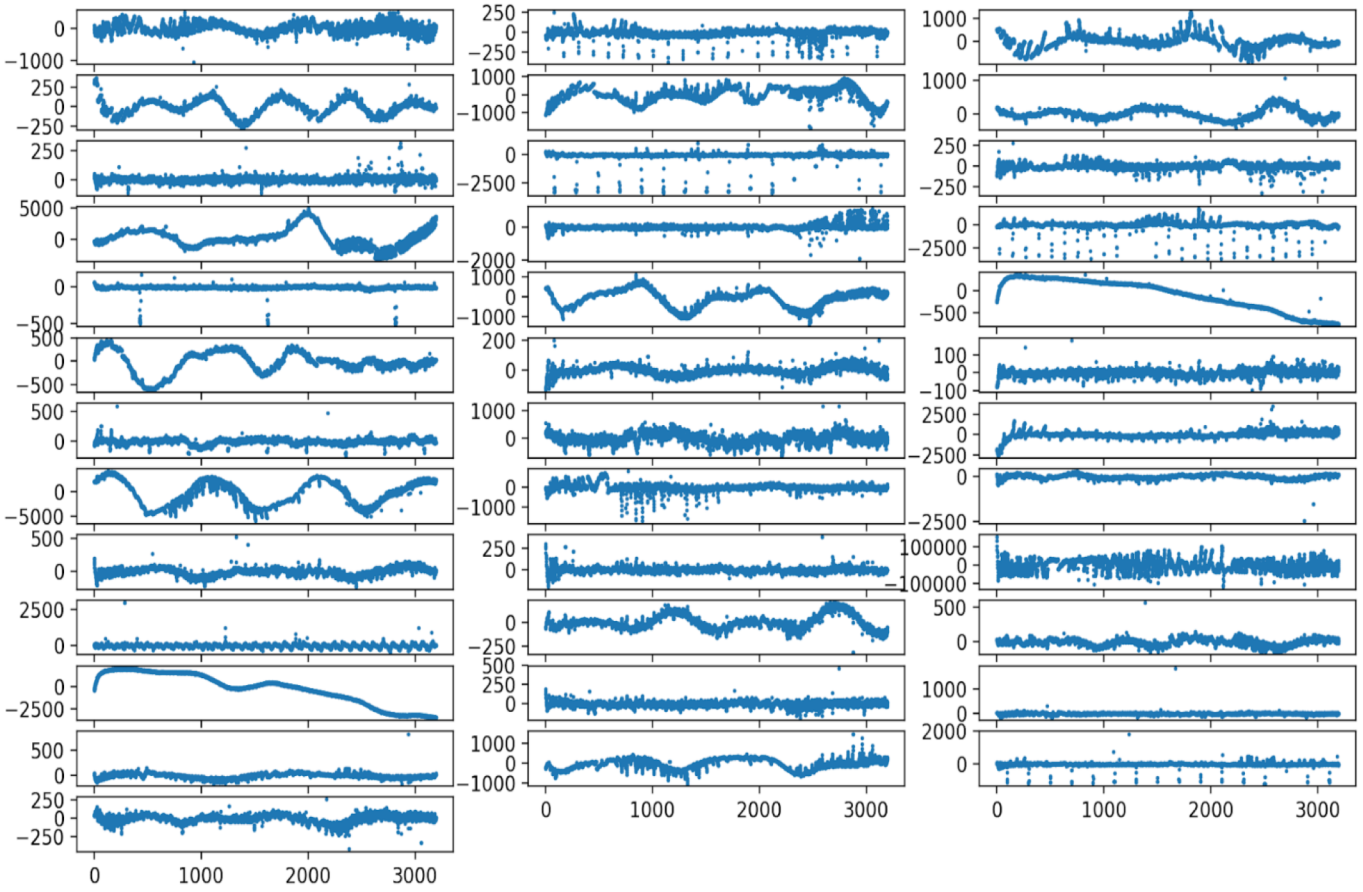
Scatter plots were generated to visualise and examine the light curve distributions.

- Large variability in the patterns emerging from samples.

Variability in the samples is largely caused by the Stellar Astrophysics specific to each star and is a big factor in the occlusion of transits in certain samples along with planet specific variables such as mass and size.

This includes:

- Pulsations
- Star spots
- Solar flares - Sharp brightness spikes


0
−1000
250
0
−250
250
0
5000
0
0
−500
500
0
−500
500
0
0
−5000
500
0
2500
0
0
−2500
500
0
250
0
−250
0
1000
2000
3000
250
0
−250
1000
0
−1000
0
−2500
0
−2000
1000
0
−1000
200
0
1000
0
0
−1000
250
0
0
−250
500
250
0
1000
0
−1000
0
1000
2000
3000
1000
0
1000
0
250
0
−250
0
−2500
0
−500
100
0
−100
2500
0
−2500
0
−2500
100000
0
−100000
500
0
1000
0
2000
0
0
1000
2000
3000

Pre-Processing (Flattening)

Processing the raw data was required to expose relevant transit features in the data.

Flattening

1. Remove missing values
2. Remove points of other planet transits in multi-planet system light curves
3. Lightcurve is split at each timestep to create “gaps”
4. Fit a piecewise polynomial spline to the light curve
    a. Optimal break-point spacings are needed to find the best fit spline for each type of lightcurve.
    b. The Bayesian Information Criterion (BIC) was used to select the best fit spline.
5. Divide the original lightcurve by the spline

The procedure results in a flattened light curve without stellar variability noise

Processing (Folding)

The flattened light curves undergo a phase folding procedure

Folding

1. Retrieve planets' **tce_period** and **bjk0** from TCE table.
2. Fold the transit period (**tce_period**) at the centre of the first detected transit (**bjk0**).

The procedure results in a folded 1d vector of the light curve with the event centered. This helps to **expose the transit** from the rest of the light curve.

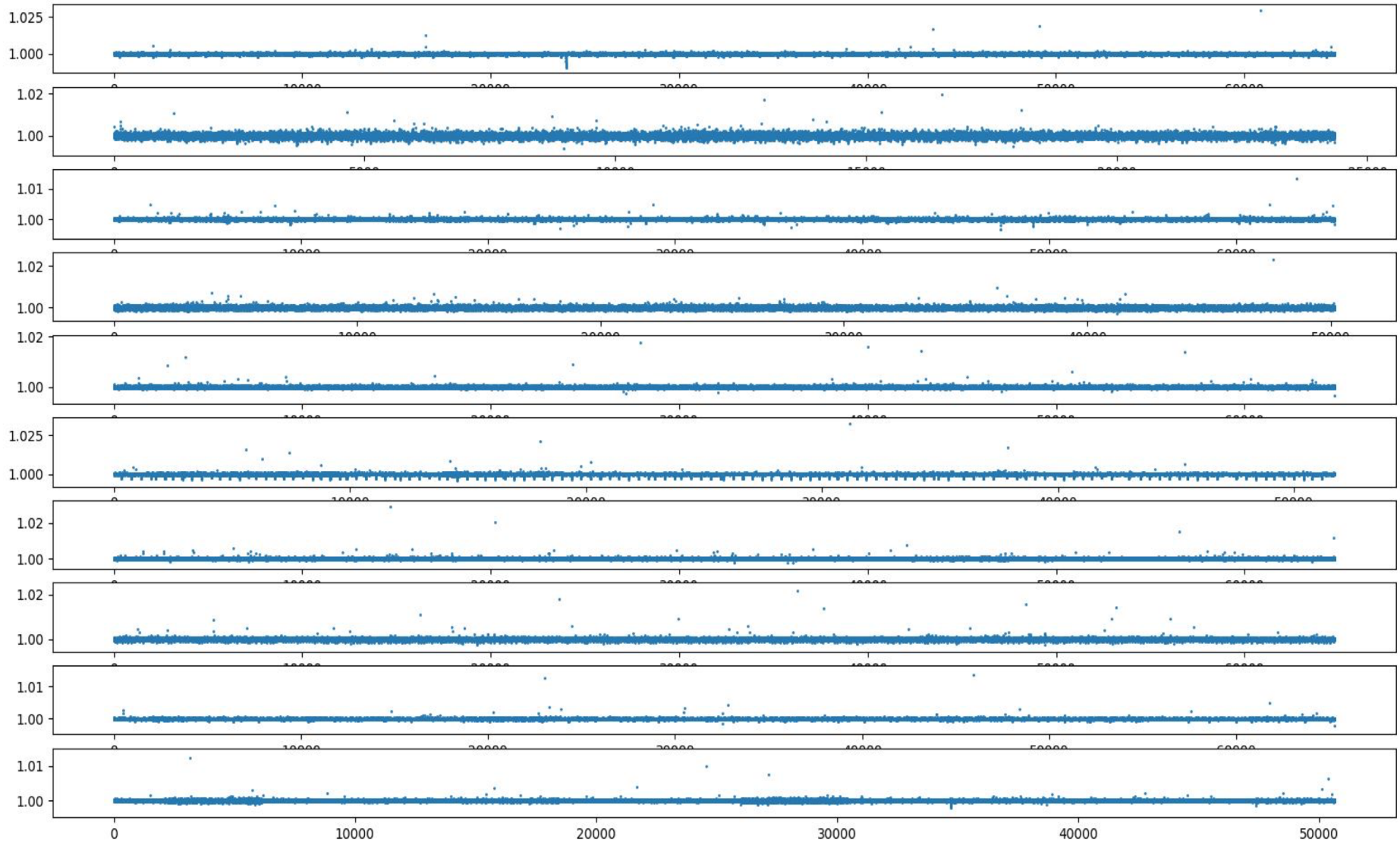
Processing (Binning)

The final processing step is to bin the folded light curves by applying a median filter to the lightcurve signal.

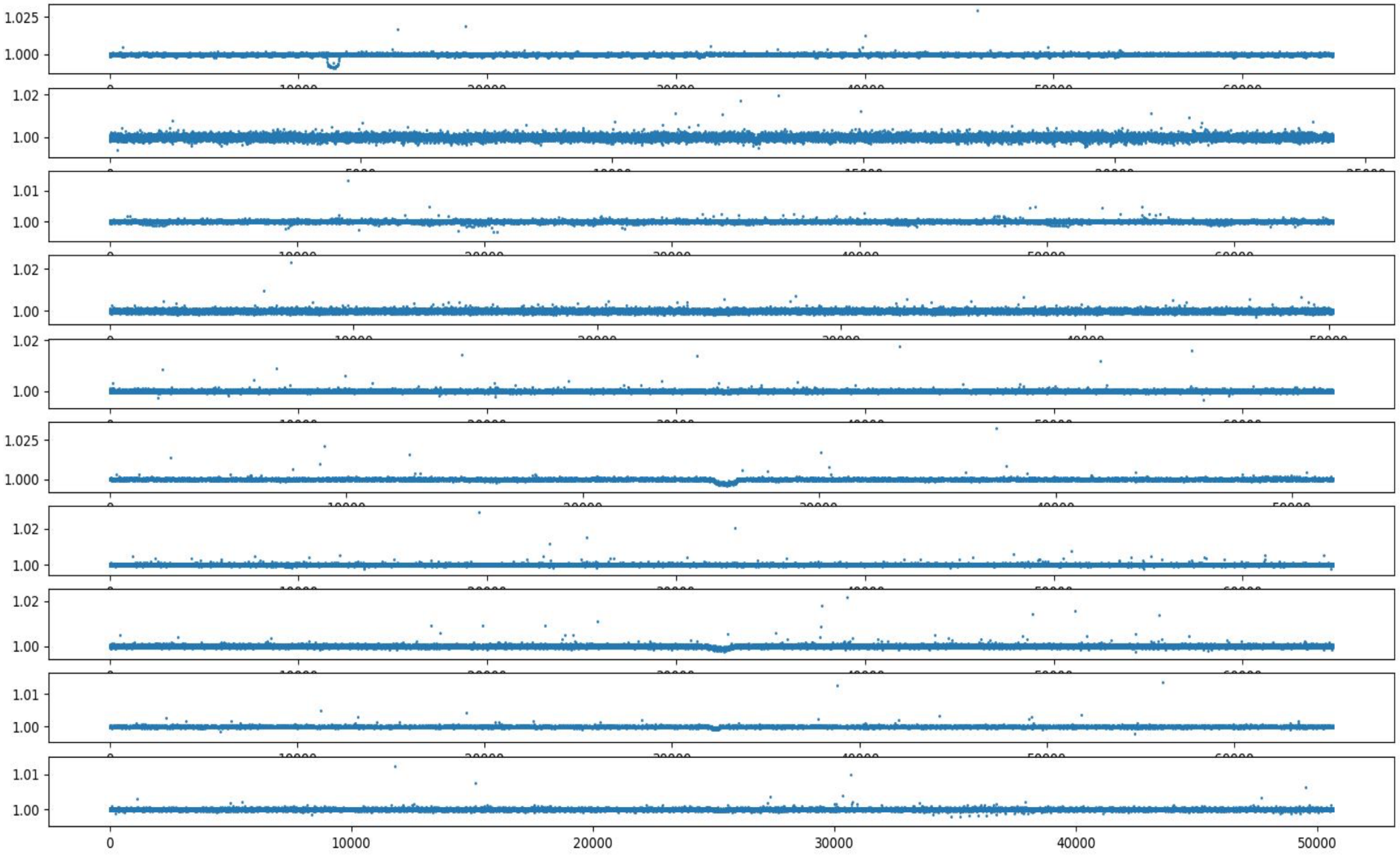
1. Define a set number of bins (2000) along the light curve axis. Each bin with a width of 1/2000.
2. Original light curve is interpolated into this range
3. The median of the flux values falling within the interval widths is used as the value of each bin.

The procedure results in a binned 1d vector for each lightcurve of size 2000, resulting in **enhanced view of the transit** due to the **massive reduction in the total number of data points** and thus, **scatter**.

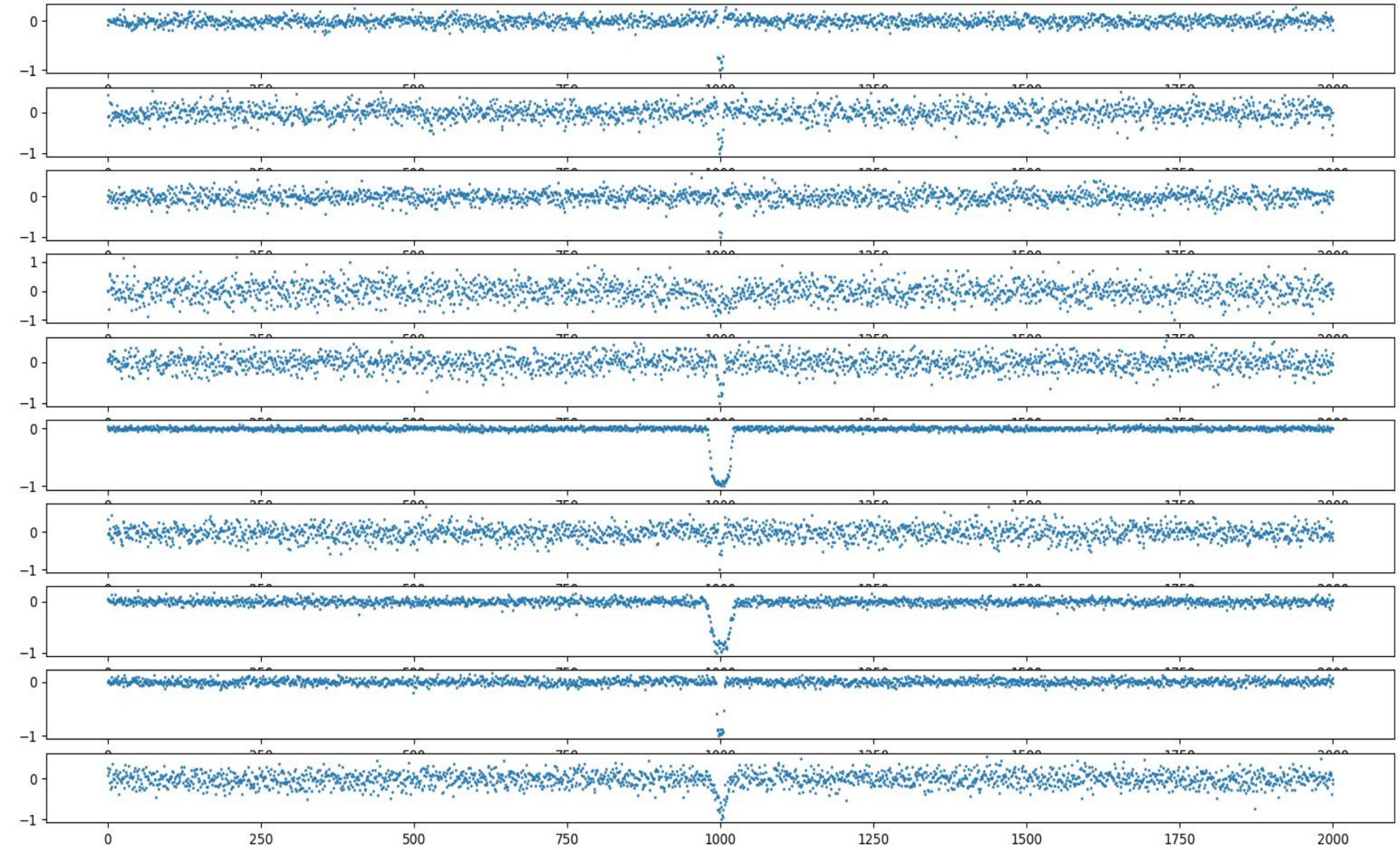
Flattened Light Curves



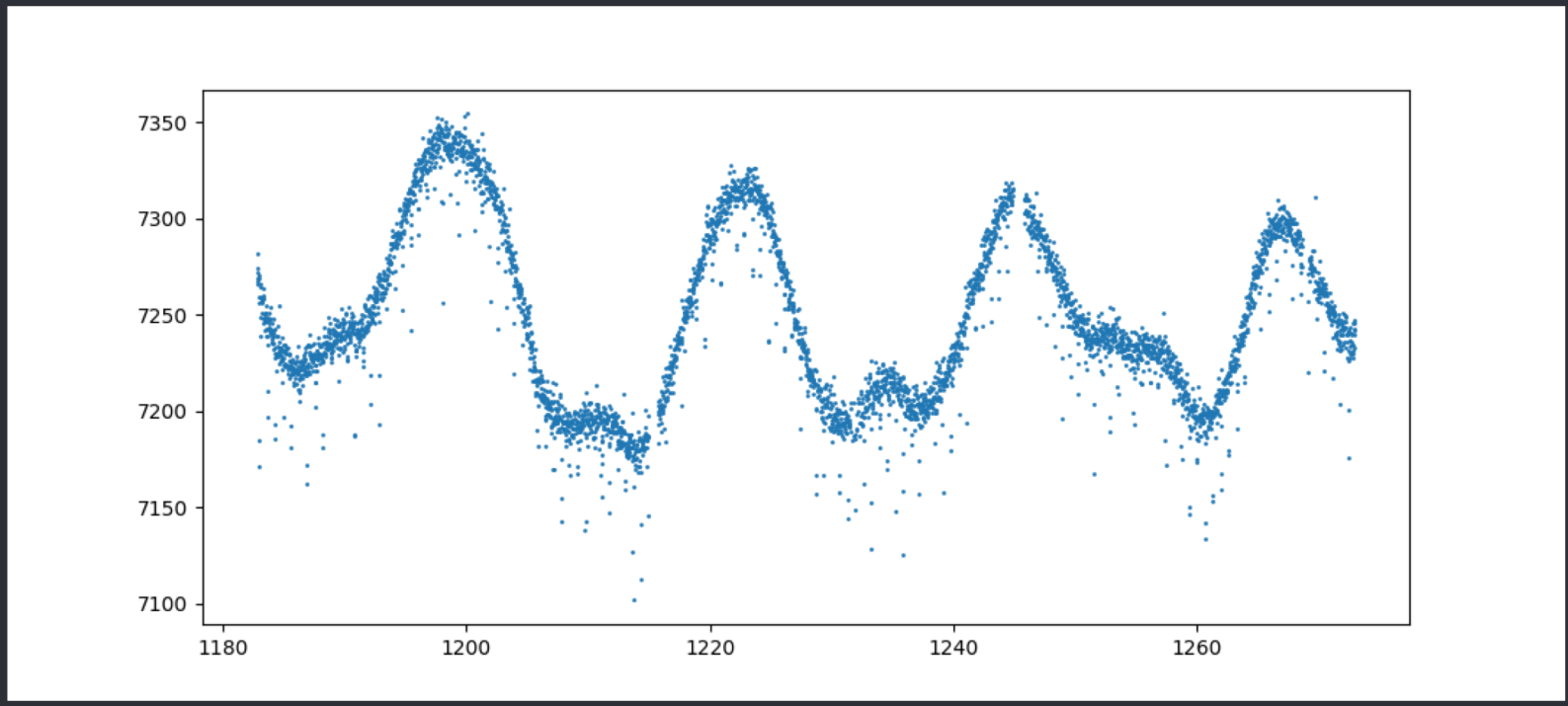
Folded Light Curves



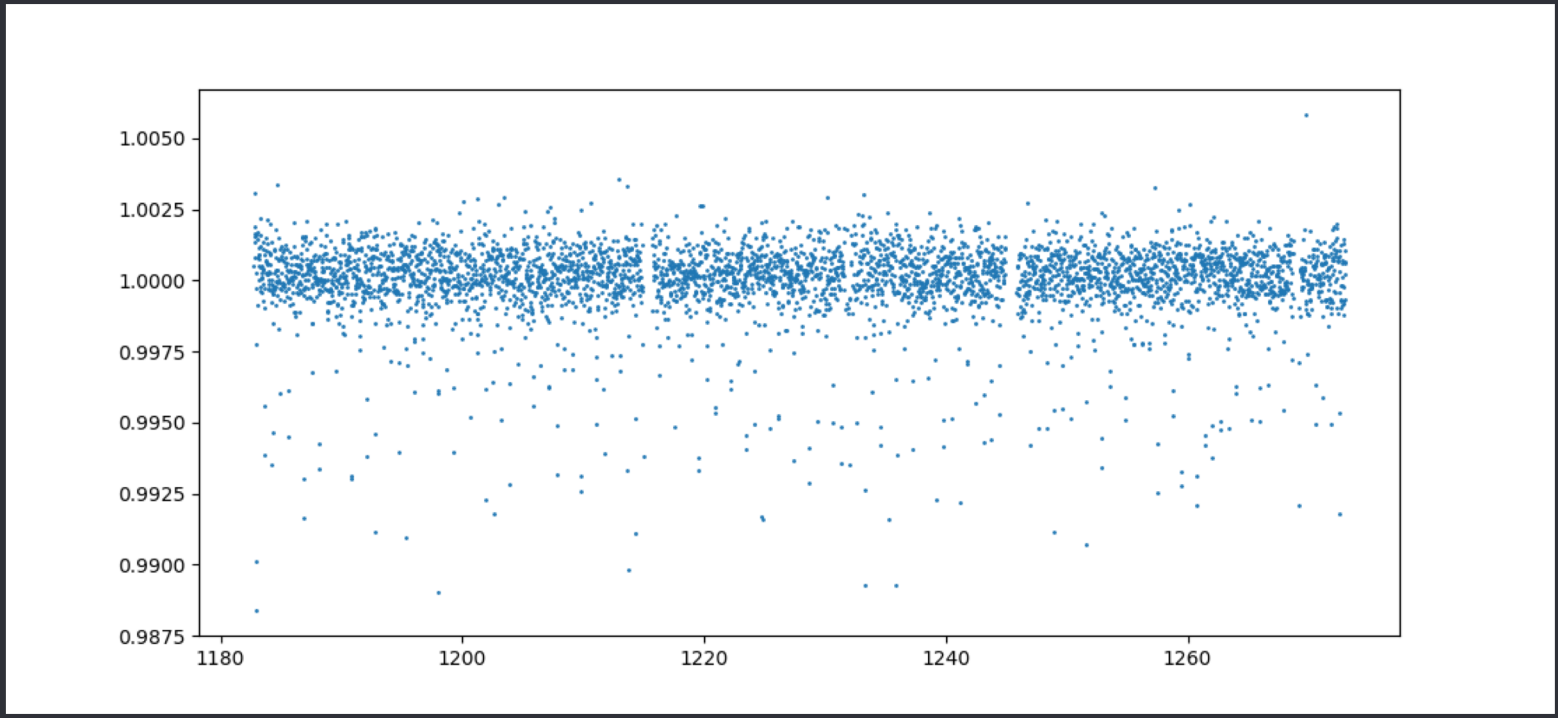
Binned Light Curves



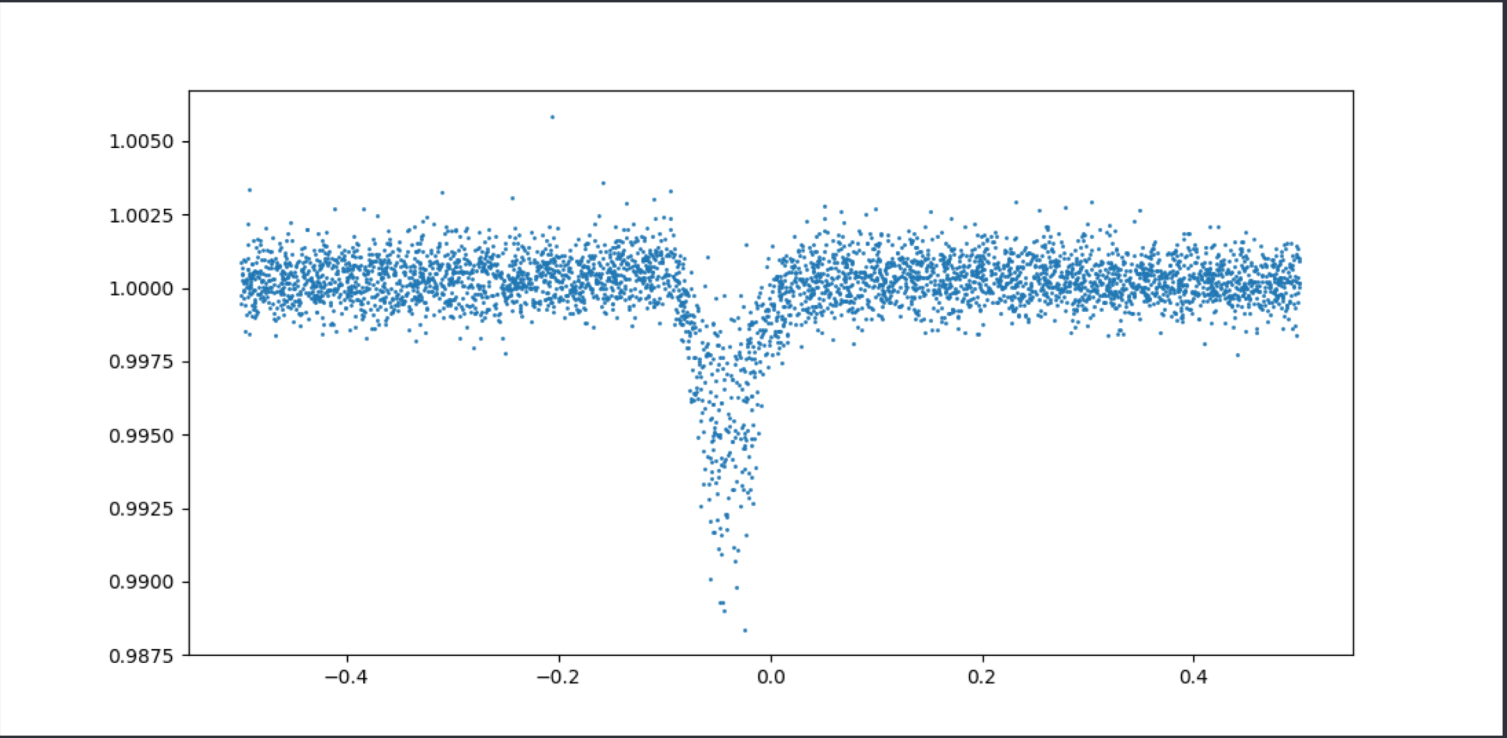
KIC 12557548 - Original PDCSAP (Using PyKe)



KIC 12557548 - Flattened (Using PyKe)



KIC12557548 - Folded (Using PyKe)



3 Modelling

Model Development

- Both CNN and LSTM model architectures were developed for the time-series binary classification task.
- Keras was used as the main deep learning framework to facilitate development.
    - Sits on top of Tensorflow
    - Good for fast prototyping
    - Easy stacking of layers of different kinds
    - Reading and saving pre-trained models
    - GPU training through CUDA
- Many other tools e.g, numpy, pandas, matplotlib, sklearn

Configuration Selection

- Selection of the best performing configuration involves the tuning of many hyperparameters.
  - Hyperopt was used to define a search space of possible evaluations of hyperparameter setups for each model.
  - An **objective function** was defined with the goal of **maximising the F1 Score** of the evaluated model configurations.
  - The search optimization algorithm used by Hyperopt is called Tree of Parzen Estimators - a type of bayesian optimization.

Hyperparameter Search Spaces

Table 4.1: CNN Parameter Search Space

Parameter	Distribution	Space
Number Blocks	Choice	[0,1,2,3]
Filters	Choice	[8,16,32,64,128,254]
Kernel Size	Choice	[3, 5, 7, 9]
Pooling Type	Choice	[Max, Average]
Pooling Size	Choice	[2,3,4]
Pooling Strides	Choice	[2,3,4]
Conv Dropout	Uniform	[0.0 - 0.35]
FC Dropout	Uniform	[0.0 - 0.6]
FC Units	Choice	[32,64,128,254]
Batch Size	Choice	[16,32]
Learning Rate Mult	Log Uniform	[-0.5 - 0.5]
Momentum	Uniform	[0.0 0.5]
Batch Normalisation	Choice	[32,64,128,254]
Number Epochs	Uniform	[10.0 - 50.0]
Activation	Choice	[ReLU, PReLU]

Table 4.2: LSTM Parameter Search Space

Parameter	Distribution	Space
Number of LSTM Layers	Choice	[0, 1]
Number of LSTM Units	Choice	[2, 5, 10, 15]
Number of Fully Connected Layers	Choice	[0, 1, 2]
Number of Fully Connected Units	Choice	[32, 64, 128]
Dropout	Uniform	[0.0 - 0.5]
Batch Size	Choice	[16, 32]
Learning Rate Mult	Log Uniform	[-0.5 - 0.5]
Momentum	Uniform	[0.0 - 0.5]
Batch Normalisation	Choice	[32, 64, 128, 254]
Number Epochs	Uniform	[10.0 - 50.0]
Activation	Choice	[ReLU, PReLU]

Evaluation Procedure

Each hyperparameter configuration needed to be **evaluated** in way that **reduced the possibility of under/over-fitting** the dataset and **prevented** the **introduction of bias** into the model-fitting procedure.

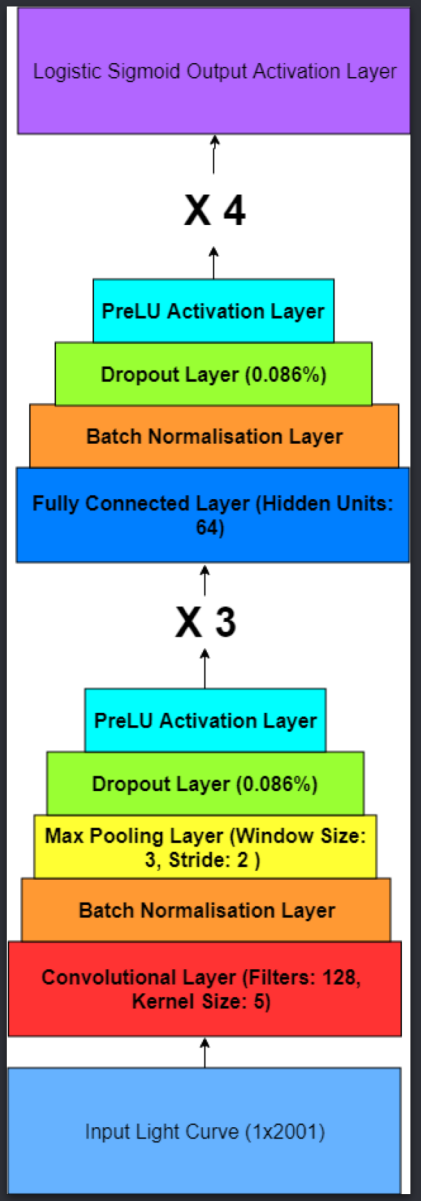
1. Dataset was split into 90% (CV Tuning) and 10% (CV Testing)
2. 5-Fold CV - Selection of hyperparameter configuration on the tuning split.
3. 5-Fold CV - Testing generalization of selected model on test split.

    Full Nested CV would have been a superior evaluation strategy, albeit, a lot more computationally expensive

Selected Topologies and Training Configurations (CNN)

CNN optimization ran for **100** evaluations

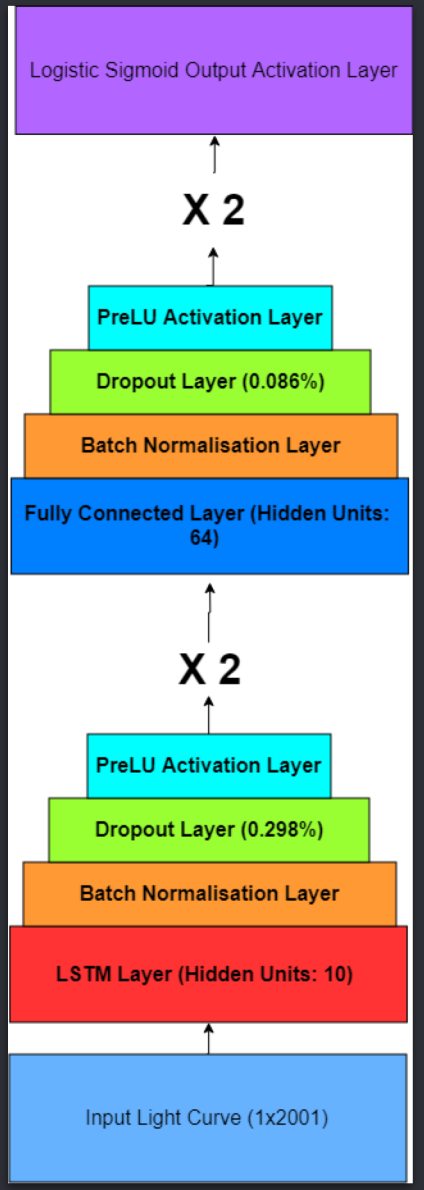
Epochs	Batch Size	Optimization	LR	Momentum	Decay
36	32	SGD	0.001345 ...	0.25	0.0001



Selected Topologies and Training Configurations (LSTM)

LSTM optimization ran for **10** evaluations

Epochs	Batch Size	Optimization	LR	Momentum	Decay
43	16	SGD	0.001643 ...	0.25	0.0001



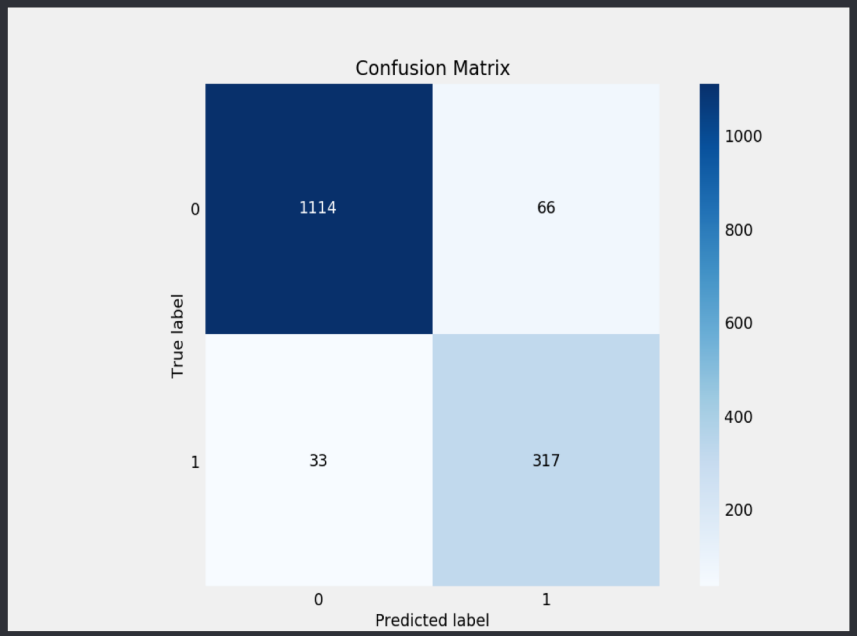
4 Results

Results

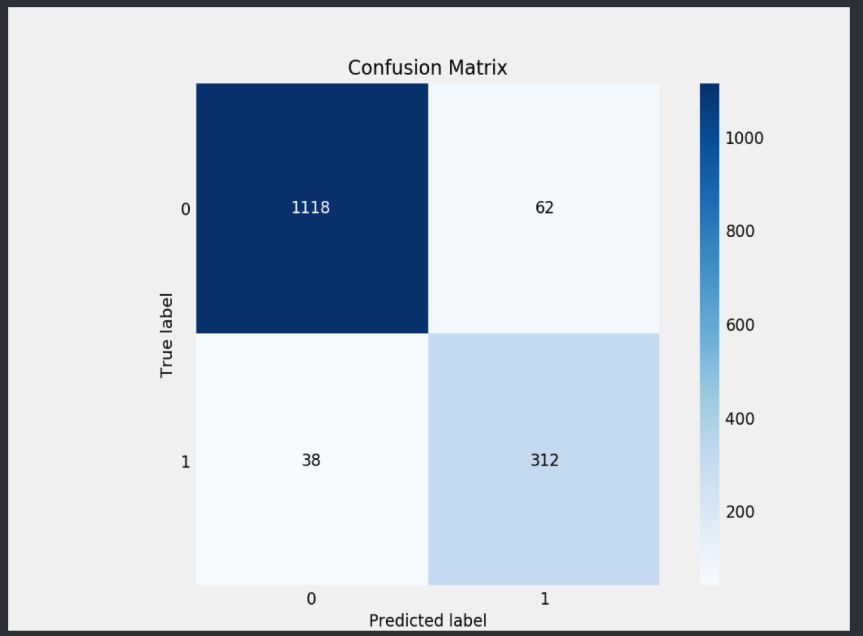
Model	Dataset	Accuracy (%)	AUC (%)	Precision (%)	Recall (%)	F1 (%)	Parameters	CV Time (5 folds)
CNN	Processed	0.93529	0.96453	0.93839	0.93529	0.93629	4.3m	49 mins
LSTM	Processed	0.93464	0.9652	0.93673	0.93464	0.93539	1.2m	3hr 30mins
Astronet (CNN)	Processed, local and global views	**0.960**	**0.988**	0.93	**0.95**	N/A	N/A	N/A
Astronet (MLP)	Processed, local and global views	**0.941**	**0.977**	N/A	**N/A**	N/A	N/A	N/A

Confusion Matrices

CNN

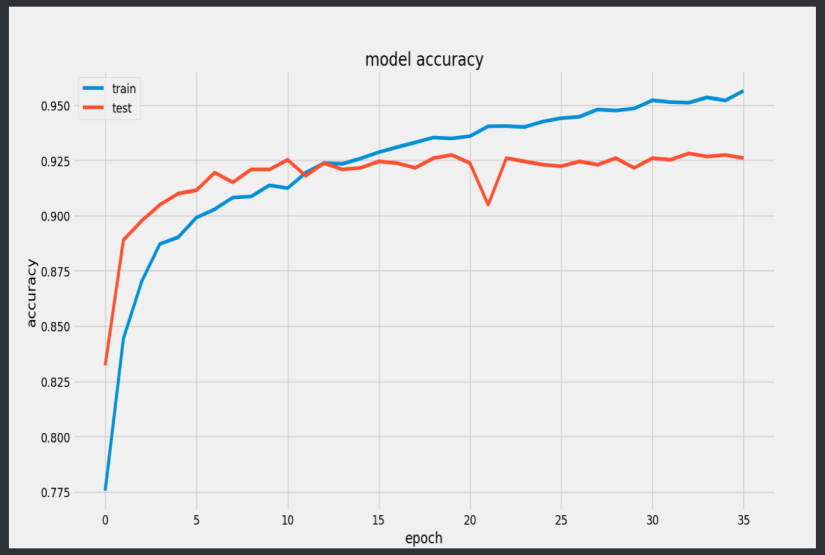


LSTM

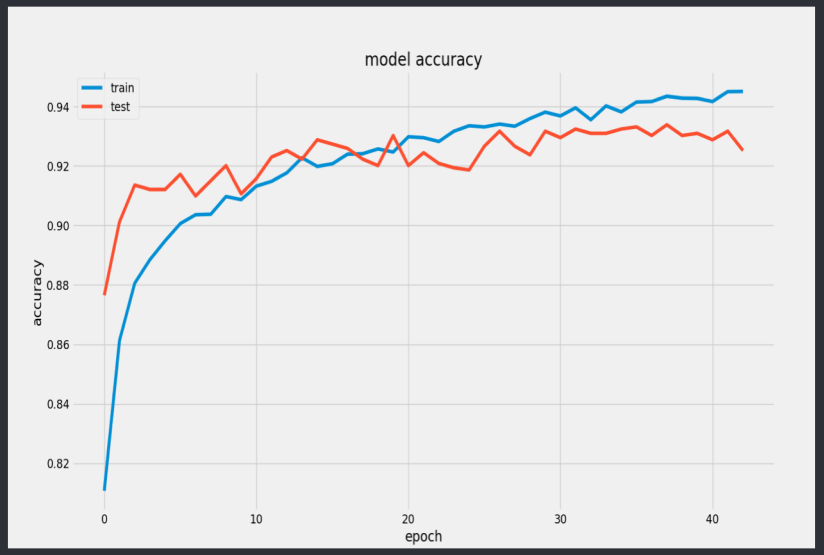


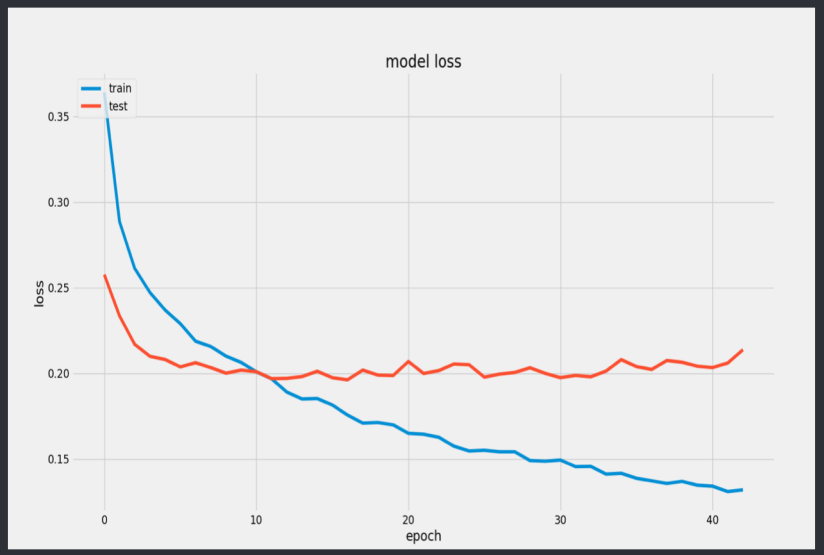
Training Performance

CNN



LSTM





5 Discussion

Comparison

- Both the LSTM and CNN architectures performed equally well on unseen test data.
  - Very different in the way each architecture models the data
    - CNNs use of convolutional and pooling layers to extract local properties and learn a representation of important regions
    - LSTMs use cell states to remember information from previous timesteps
- Both ideal for modelling the variability in the light curve transits.
  - No feature engineering required and only mandatory pre-processing was applied to the data
  - LSTM use is novel in this particular problem
- LSTMs are less optimized for parallel computation using a GPU
  - Much longer training times than CNN even with less internal training parameters

Improvements

- Seeking knowledge directly from experts
- Multiple views of the input data
- Generate synthetic data for training
  - Oversampling (INOS)
  - Transit generation
- Computational resources
  - More search optimization evaluations
    - Better configuration found
  - Larger hyperparameter search space
    - Higher complexity model architectures could be evaluated
- Evaluation
  - Baseline approach e.g. BLS
    - Better measure of how much better the CNN and LSTM architectures performed
  - Testing on synthetic data produced by NASA to tune the Kepler pipeline
    - Better comparison to real state of art systems
  - Full implementation of Nested CV
    - Improved generalization assessment of models
  - Visualisation techniques

Research Findings

- Deep learning potential in Astronomy:
  - Ability to produce accurate predictions with high confidence most of the time
  - Much quicker at reviewing new data and producing new predictions compared to a field expert, while sacrificing reliability (for now).
  - Flexible enough to be trained with more data
- Their black-box nature could be part of the reason that systems such as Robovetter are still prefered in real systems such as Kepler.
  - Manual reviews of the model predictions should still be conducted.
- Ultimately, given more time, deep learning approaches are bound to become state of the art and used extensively in this domain, especially as neural network visualisation techniques advance.

ANY QUESTIONS?